

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-181878

(43)Date of publication of application : 30.06.2000

(51)Int.Cl. G06F 15/16  
G06F 9/38  
G06F 17/16

(21)Application number : 10-375410

(71)Applicant : NEC CORP

(22)Date of filing : 15.12.1998

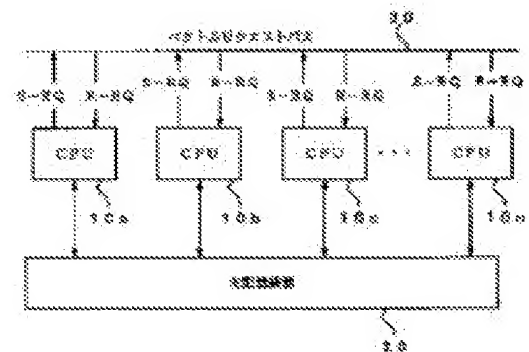
(72)Inventor : NAKAZATO SATOSHI

## (54) COMMON MEMORY TYPE VECTOR PROCESSING SYSTEM, ITS CONTROL METHOD AND STORAGE MEDIUM STORED WITH CONTROL PROGRAM FOR VECTOR PROCESSING

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To flexibly change the number of vector pipelines attached to a scalar processing means according to application purposes by allowing vector processing means of slave CPUs to operate as multiple vector pipelines adjunctly to the scalar processing means of a master CPU.

**SOLUTION:** CPUs 10a to 10n starting a single main storage device 20 constitute a common memory type parallel processing system. The CPUs 10a to 10n are interconnected through a vector request bus 30. Some of the CPUs 10a to 10n are set as master CPUs and others as slave CPUs. The master CPUs perform scalar processing and issue vector processing instructions to other CPUs. The slave CPUs operate as multiple vector pipelines synchronously with the vector processing parts in the master CPUs. At this moment, the scalar processing parts of the slave CPUs enter into halt states and their vector processing parts function effectively.



(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-181878

(P2000-181878A)

(43) 公開日 平成12年6月30日 (2000.6.30)

(51) Int.Cl.<sup>7</sup>

識別記号

F I

テーマコード\* (参考)

G 0 6 F 15/16

6 1 0

G 0 6 F 15/16

6 1 0 F 5 B 0 1 3

9/38

3 1 0

9/38

3 1 0 G 5 B 0 4 5

17/16

15/347

F 5 B 0 5 6

審査請求 有 請求項の数28 F D (全 14 頁)

(21) 出願番号

特願平10-375410

(22) 出願日

平成10年12月15日 (1998. 12. 15)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 中里 聡

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100093595

弁理士 松本 正夫

Fターム(参考) 5B013 AA14

5B045 BB04 BB48 GG09 GG17 KK08

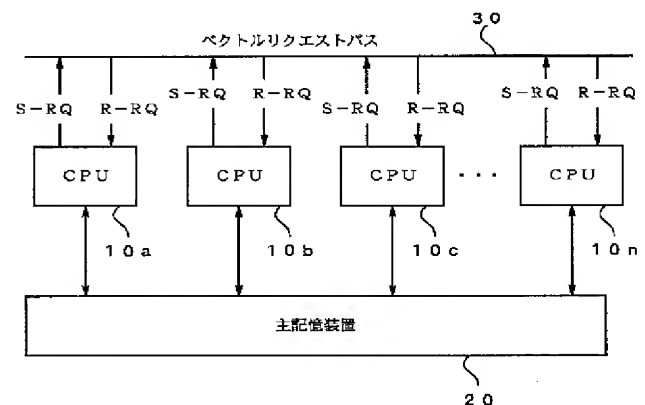
5B056 BB32 BB46 FF00 FF05 FF16

(54) 【発明の名称】 共有メモリ型ベクトル処理システムとその制御方法及びベクトル処理の制御プログラムを格納する記憶媒体

(57) 【要約】

【課題】 スカラ処理部に付随するベクトルパイプライン数を用途に応じて柔軟に変更できるベクトル処理システムを提供する。

【解決手段】 この共有メモリ型ベクトル処理システムは、各CPU 10a~10n相互を、各CPUから生成するベクトル処理命令を各CPUに転送するためのバス30によって接続し、各CPUを、ベクトル処理命令を他のCPUに対して発行するマスターCPUと、マスターCPU内のベクトル処理部と同期して多重ベクトルパイプラインとして動作するスレーブCPUとに分け、マスターCPUは、発行元のCPUを識別する発行元CPU情報を付加した前記ベクトル処理命令を発行し、バスを介して自CPUを含む全てのCPUに対して転送するメモリアクセス制御部12を備え、マスターCPUとスレーブCPUは、ベクトル処理命令に含まれる発行元CPU情報と自CPUに設定されたマスターCPU情報とを比較し、一致する場合にベクトル処理命令に基づく命令発行を行ない、一致しない場合にベクトル処理命令の無効化を行なうベクトル処理命令制御部13を備える。



## 【特許請求の範囲】

【請求項 1】 主記憶メモリを共有する複数の CPU を備え、各 CPU がスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記 CPU をマスター CPU と、複数のスレーブ CPU とに分けて設定し、前記マスター CPU の前記スカラ処理手段に付随して、前記複数のスレーブ CPU の前記ベクトル処理手段を多重ベクトルパイプラインとして動作させることを特徴とする共有メモリ型ベクトル処理システム。

【請求項 2】 主記憶メモリを共有する複数の CPU を備え、各 CPU がスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、複数の前記 CPU の前記ベクトル処理手段を全てまとめて単一の多重ベクトルパイプラインと見なし、前記各 CPU の各スカラ処理手段から前記単一の多重ベクトルパイプラインを共有しているように動作させることを特徴とするベクトル処理システム。

【請求項 3】 主記憶メモリを共有する複数の CPU を備え、各 CPU がスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記各 CPU 相互を、前記各 CPU から生成するベクトル処理命令を各 CPU に転送するためのパスによって接続し、前記各 CPU は、発行元の CPU を識別する発行元 CPU 情報を付加したベクトル処理命令を発行し、前記パスを介して自 CPU を含む全ての CPU に対して転送する発行手段と、転送された前記ベクトル処理命令に含まれる前記発行元 CPU 情報と、自 CPU のベクトルパイプラインに関する構成情報とを比較し、転送された前記ベクトル処理命令に対する処理内容を決定するベクトル処理命令制御手段とを備えることを特徴とする共有メモリ型ベクトル処理システム。

【請求項 4】 主記憶メモリを共有する複数の CPU を備え、各 CPU がスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記各 CPU 相互を、前記各 CPU から生成するベクトル処理命令を各 CPU に転送するためのパスによって接続し、前記各 CPU を、スカラ処理を実行すると共に、ベクトル処理命令を他の CPU に対して発行するマスター CPU と、前記マスター CPU から転送されてきたベクトル処理命令を受け取り、マスター CPU 内のベクトル処理部と同期して多重ベクトルパイプラインとして動作するスレーブ CPU とに分けて設定し、

10

20

30

40

50

前記マスター CPU は、発行元の CPU を識別する発行元 CPU 情報を付加した前記ベクトル処理命令を発行し、前記パスを介して自 CPU を含む全ての CPU に対して転送する発行手段を備え、前記マスター CPU とスレーブ CPU は、転送された前記ベクトル処理命令に含まれる前記発行元 CPU 情報と自 CPU に設定されたマスター CPU 情報とを比較し、一致する場合に転送された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、一致しない場合に前記ベクトル処理命令の無効化を行なうベクトル処理命令制御手段を備えることを特徴とする共有メモリ型ベクトル処理システム。

【請求項 5】 前記マスター CPU とスレーブ CPU の前記ベクトル処理命令制御手段は、転送された前記ベクトル処理命令に含まれる前記発行元 CPU 情報と、自 CPU に設定されたマスター CPU 情報とを比較する比較手段と、転送された前記ベクトル処理命令を格納する命令スタックと、前記比較手段による比較結果が一致の場合、前記ベクトル処理命令を命令スタックに格納し、比較結果が不一致の場合、前記前記ベクトル処理命令を無効化する無効化処理手段と、前記ベクトル処理部のリソース状態に基づいて、前記命令スタックに格納された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なう命令発行処理手段とを備えることを特徴とする請求項 4 に記載の共有メモリ型ベクトル処理システム。

【請求項 6】 前記マスター CPU とスレーブ CPU の前記ベクトル処理命令制御手段は、転送された前記ベクトル処理命令を前記ベクトル処理命令自体と前記発行元 CPU 情報に分離して出力する抽出手段と、分離された前記発行元 CPU 情報と、自 CPU に設定されたマスター CPU 情報とを比較する比較手段と、前記比較手段による比較結果が一致の場合、前記抽出手段からの前記ベクトル処理命令を命令スタックに格納し、比較結果が不一致の場合、前記前記ベクトル処理命令を無効化する無効化処理手段と、前記ベクトル処理部のリソース状態に基づいて、前記命令スタックに格納された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なう命令発行処理手段とを備えることを特徴とする請求項 4 に記載の共有メモリ型ベクトル処理システム。

【請求項 7】 前記マスター CPU とスレーブ CPU の前記ベクトル処理命令制御手段は、転送された前記ベクトル処理命令を格納する命令スタックと、前記命令スタックに格納された前記ベクトル処理命令に

含まれる前記発行元CPU情報と、自CPUに設定されたマスターCPU情報とを比較し、比較結果が一致の場合、前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、比較結果が不一致の場合、前記ベクトル処理命令に基づく命令発行を抑止し前記命令スタックの該当エリアを解放する命令発行処理手段とを備えることを特徴とする請求項4に記載の共有メモリ型ベクトル処理システム。

【請求項8】 前記命令スタックには、前記発行元CPU情報を含んだ状態で前記ベクトル処理命令を格納することを特徴とする請求項7に記載の共有メモリ型ベクトル処理システム。

【請求項9】 前記マスターCPUとスレーブCPUの前記ベクトル処理命令制御手段は、自CPUがスレーブCPUである場合に、自CPUの前記スカラ処理手段の動作を停止させる停止手段を備えることを特徴とする請求項3乃至請求項8に記載の共有メモリ型ベクトル処理システム。

【請求項10】 前記停止手段は、自CPUに対するマスターCPUと機能するCPUを示すマスターCPU情報と、自CPUのCPU情報を格納する格納手段と、前記ベクトル処理命令が転送された際に、前記格納手段に格納されたマスターCPU情報と自CPU情報とを比較し、不一致の場合に、自CPUの前記スカラ処理手段の動作停止信号を出力する比較手段とで構成されることを特徴とする請求項9に記載の共有メモリ型ベクトル処理システム。

【請求項11】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記各CPU相互を、前記各CPUから生成するベクトル処理命令を各CPUに転送するためのパスによって接続し、前記各CPUは、発行元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、前記パスを介して自CPUを含む全てのCPUに対して転送する発行手段と、転送された前記ベクトル処理命令を、前記発行元CPU情報に基づいて各CPU毎に対応した複数の命令スタックに格納し、前記複数の命令スタック毎の優先順位と前記ベクトル処理手段のリソース情報に基づいて、前記ベクトル処理命令に基づく命令発行を制御するベクトル処理命令制御手段とを備えることを特徴とする共有メモリ型ベクトル処理システム。

【請求項12】 前記各CPUのベクトル処理命令制御手段は、各CPU毎に対応した複数の命令スタックと、転送された前記ベクトル処理命令に含まれる前記発行元

CPU情報を検出し、前記ベクトル処理命令を対応する前記命令スタックに格納する命令発行元検出手段と、複数の前記複数の命令スタック毎に、何れの命令スタックのベクトル処理命令に基づく命令発行を優先するか決定する調停手段と、

前記調停手段による決定内容と前記ベクトル処理手段のリソース情報に基づいて、前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なう命令発行処理手段とを備えることを特徴とする請求項11に記載の共有メモリ型ベクトル処理システム。

【請求項13】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記CPUをマスターCPUと、複数のスレーブCPUとし、

前記マスターCPUの前記スカラ処理手段に付随して、前記複数のスレーブCPUの前記ベクトル処理手段を多重ベクトルパイプラインとして動作させることを特徴とする共有メモリ型ベクトル処理システムの制御方法。

【請求項14】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、複数の前記CPUの前記ベクトル処理手段を全てまとめて単一の多重ベクトルパイプラインと見なし、前記各CPUの各スカラ処理手段から前記単一の多重ベクトルパイプラインを共有しているように動作させることを特徴とする共有メモリ型ベクトル処理システムの制御方法。

【請求項15】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、

前記各CPUにおいて、発行元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、相互に接続されたパスを介して自CPUを含む全てのCPUに対して転送し、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUのベクトルパイプラインに関する構成情報とを比較し、転送された前記ベクトル処理命令に対する処理内容を決定することを特徴とする共有メモリ型ベクトル処理システムの制御方法。

【請求項16】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、

前記各CPUを、スカラ処理を実行すると共に、ベクトル処理命令を他のCPUに対して発行するマスターCPUと、前記マスターCPUから転送されてきたベクトル処理命令を受け取り、マスターCPU内のベクトル処理

部と同期して多重ベクトルパイプラインとして動作するスレーブCPUとに分けて設定し、前記マスターCPUにおいて、発行元のCPUを識別する発行元CPU情報を付加した前記ベクトル処理命令を発行し、相互に接続されたパスを介して自CPUを含む全てのCPUに対して転送し、前記マスターCPUとスレーブCPUにおいて、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と自CPUに設定されたマスターCPU情報とを比較し、比較結果が一致する場合に転送された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、一致しない場合に前記ベクトル処理命令の無効化を行なうことを特徴とする共有メモリ型ベクトル処理システムの制御方法。

【請求項17】 前記マスターCPUとスレーブCPUにおいて、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUに設定されたマスターCPU情報とを比較し、比較結果が一致の場合、前記ベクトル処理命令を命令スタックに格納し、比較結果が不一致の場合、前記前記ベクトル処理命令を無効化し、前記ベクトル処理手段のリソース状態に基づいて、前記命令スタックに格納された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なうことを特徴とする請求項16に記載の共有メモリ型ベクトル処理システムの制御方法。

【請求項18】 前記マスターCPUとスレーブCPUにおいて、命令スタックに格納された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUに設定されたマスターCPU情報とを比較し、比較結果が一致の場合、前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、比較結果が不一致の場合、前記ベクトル処理命令に基づく命令発行を抑止し前記命令スタックの該当エリアを解放することを特徴とする請求項17に記載の共有メモリ型ベクトル処理システムの制御方法。

【請求項19】 自CPUがスレーブCPUである場合に、自CPUの前記スカラ処理手段の動作を停止させることを特徴とする請求項15乃至請求項18に記載の共有メモリ型ベクトル処理システムの制御方法。

【請求項20】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記各CPU相互を、前記各CPUから生成するベクトル処理命令を各CPUに転送するためのパスによって接続し、

前記各CPUにおいて、発行元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、前記パスを介して自CPUを含む全てのCPUに対して転送し、転送された前記ベクトル処理命令を、前記発行元CPU情報に基づいて各CPU毎に対応した複数の命令スタックに格納し、前記複数の命令スタック毎の優先順位と前記ベクトル処理手段のリソース情報に基づいて、前記ベクトル処理命令に基づく命令発行を制御することを特徴とする共有メモリ型ベクトル処理システムの制御方法。

10

【請求項21】 前記各CPUにおいて、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報を検出し、前記ベクトル処理命令に対応する命令スタックに格納し、複数の前記複数の命令スタック毎に、何れの命令スタックのベクトル処理命令に基づく命令発行を優先するか決定し、前記決定内容と前記ベクトル処理手段のリソース情報に基づいて、前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なうことを特徴とする請求項20に記載の共有メモリ型ベクトル処理システムの制御方法。

20

【請求項22】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムの制御を行なう制御プログラムを格納する記憶媒体において、前記制御プログラムは、発行元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、相互に接続されたパスを介して自CPUを含む全てのCPUに対して転送し、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUのベクトルパイプラインに関する構成情報とを比較し、転送された前記ベクトル処理命令に対する処理内容を決定することを特徴とする共有メモリ型ベクトル処理システムの制御プログラムを格納する記憶媒体。

30

【請求項23】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムの制御を行なう制御プログラムを格納する記憶媒体において、前記制御プログラムは、

40

前記各CPUを、スカラ処理を実行すると共に、ベクトル処理命令を他のCPUに対して発行するマスターCPUと、前記マスターCPUから転送されてきたベクトル処理命令を受け取り、マスターCPU内のベクトル処理部と同期して多重ベクトルパイプラインとして動作するスレーブCPUとに分けて設定し、

50

前記マスターCPUにおいて、発行元のCPUを識別する発行元CPU情報を付加した前記ベクトル処理命令を発行し、前記パスを介して自C

PUを含む全てのCPUに対して転送し、前記マスターCPUとスレーブCPUにおいて、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と自CPUに設定されたマスターCPU情報とを比較し、比較結果が一致する場合に転送された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、一致しない場合に前記ベクトル処理命令の無効化を行なうことを特徴とする共有メモリ型ベクトル処理システムの制御プログラムを格納する記憶媒体。

【請求項24】 前記制御プログラムは、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUに設定されたマスターCPU情報とを比較し、比較結果が一致の場合、前記ベクトル処理命令を命令スタックに格納し、比較結果が不一致の場合、前記前記ベクトル処理命令を無効化し、前記ベクトル処理手段のリソース状態に基づいて、前記命令スタックに格納された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なうことを特徴とする請求項23に記載の共有メモリ型ベクトル処理システムの制御プログラムを格納する記憶媒体。

【請求項25】 前記制御プログラムは、命令スタックに格納された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUに設定されたマスターCPU情報とを比較し、比較結果が一致の場合、前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、比較結果が不一致の場合、前記ベクトル処理命令に基づく命令発行を抑止し前記命令スタックの該当エリアを解放することを特徴とする請求項23に記載の共有メモリ型ベクトル処理システムの制御プログラムを格納する記憶媒体。

【請求項26】 前記制御プログラムは、自CPUがスレーブCPUである場合に、自CPUの前記スカラ処理手段の動作を停止させることを特徴とする請求項22乃至請求項25に記載の共有メモリ型ベクトル処理システムの制御プログラムを格納する記憶媒体。

【請求項27】 主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムの制御を行なう制御プログラムを格納する記憶媒体において、前記制御プログラムは、発行元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、相互に接続されるパスを介して自CPUを含む全てのCPUに対して転送し、転送された前記ベクトル処理命令を、前記発行元CPU情報に基づいて各CPU毎に対応した複数の命令スタックに格納し、前記複数の命令スタック毎の優先順位と前記ベクトル処理手段のリソース情報に基づいて、前記ベ

クトル処理命令に基づく命令発行を制御することを特徴とする共有メモリ型ベクトル処理システムの制御プログラムを格納する記憶媒体。

【請求項28】 前記制御プログラムは、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報を検出し、前記ベクトル処理命令に対応する命令スタックに格納し、複数の前記複数の命令スタック毎に、何れの命令スタックのベクトル処理命令に基づく命令発行を優先するか決定し、

前記決定内容と前記ベクトル処理手段のリソース情報に基づいて、前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なうことを特徴とする請求項27に記載の共有メモリ型ベクトル処理システムの制御プログラムを格納する記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理部と複数のベクトルパイプラインを構成するベクトル処理部を有してなる共有メモリ型ベクトル処理システムに関する。

【0002】

【従来の技術】図9に、従来のベクトル処理装置におけるCPUを用いた共有メモリ型並列処理システムの構成を示す。このシステムでは、複数のCPU100a～100nが1つの主記憶装置200を共有して接続されている。

【0003】各CPU100a～100nの詳細構成図を図10に示す。各CPU100a～100nは、図示のように、スカラ処理部101、命令制御部102、ベクトル処理部104a～104n及びメモリアクセスネットワーク部105を備えて構成される。

【0004】スカラ処理部101から発行された外部処理命令「EX-RQ」は、命令制御部102へと転送される。命令制御部102では、自CPU内部にのみ存在するベクトル処理部104a～104nのリソースを管理することによりベクトル処理命令「V-RQ」を発行する。

【0005】従って、各CPU100a～100n内部におけるスカラ処理部101とベクトルパイプラインとの構成は、常に一定であり変更することができない。

【0006】従来のベクトル処理装置の例が例えば特開昭63-127368号及び特開昭63-10263号等に開示されている。これらの公報に開示されるベクトル処理装置においては、何れもスカラ処理部とベクトルパイプラインとの構成が一定であり、スカラ処理部に付随するベクトルパイプライン数を用途に応じて柔軟に変更できる構成とはなっていない。

【0007】

【発明が解決しようとする課題】 上述した従来のベクトル処理装置においては、以下のような問題点がある。

【0008】 第1の問題点は、実行するアプリケーションによってベクトル化率等が異なるのに対して、適切なベクトル処理リソースを割り当てられないことである。

【0009】 その理由は、各CPUにおけるベクトルパイプライン数は常に一定であるため、想定しているよりも低いベクトル化率のアプリケーションを実行した場合、ベクトルリソースの余剰が発生してしまう。また逆に、より高いベクトル化率やより長いベクトル長のアプリケーションを実行しても、予め固定的に構成されたベクトルパイプライン構成によりベクトル処理性能の上限が固定されているため、更なる処理性能向上を行なうことができないからである。

【0010】 第2の問題点は、LSIの集積度が向上してもスカラ処理部とベクトルパイプラインとを別のLSIとして開発する必要があることである。

【0011】 その理由は、LSIの集積度が向上し、スカラ処理部とベクトルパイプラインの1本分程度が1チップ化できるようになってきたが、従来の多重ベクトルパイプライン構成では、このようなLSIを複数接続する際に各LSIに存在するスカラ処理部は全く利用できずハードウェア量を無駄に使用することになるため、従来通りスカラ処理部とベクトルパイプラインとを別のLSIとして開発することになってしまう。ところが、この方法ではLSIの開発工数が増大すること、LSIの開発品種数が増加すること、1種のLSI当たりの生産個数が減少することなどコスト増となる項目が多くなる。

【0012】 本発明の目的は、スカラ処理部に付随するベクトルパイプライン数を用途に応じて柔軟に変更できるベクトル処理システムを提供することにある。

【0013】 本発明の他の目的は、独立する各プロセッサのスカラ処理部からの単一のベクトルパイプラインを共有しているように動作するベクトル処理システムを提供することにある。

【0014】

【課題を解決するための手段】 上記目的を達成する請求項1の本発明は、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記CPUをマスターCPUと、複数のスレーブCPUとに分けて設定し、前記マスターCPUの前記スカラ処理手段に付随して、前記複数のスレーブCPUの前記ベクトル処理手段を多重ベクトルパイプラインとして動作させることを特徴とする。

【0015】 請求項2の本発明では、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、複数の前記CPUの前記ベクトル

ル処理手段を全てまとめて単一の多重ベクトルパイプラインと見なし、前記各CPUの各スカラ処理手段から前記単一の多重ベクトルパイプラインを共有しているように動作させることを特徴とする。

【0016】 請求項3の本発明では、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記各CPU相互を、前記各CPUから生成するベクトル処理命令を各CPUに転送するためのバスによって接続し、前記各CPUは、発行元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、前記バスを介して自CPUを含む全てのCPUに対して転送する発行手段と、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUのベクトルパイプラインに関する構成情報とを比較し、転送された前記ベクトル処理命令に対する処理内容を決定するベクトル処理命令制御手段とを備えることを特徴とする。

【0017】 請求項4の本発明では、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記各CPU相互を、前記各CPUから生成するベクトル処理命令を各CPUに転送するためのバスによって接続し、前記各CPUを、スカラ処理を実行すると共に、ベクトル処理命令を他のCPUに対して発行するマスターCPUと、前記マスターCPUから転送されてきたベクトル処理命令を受け取り、マスターCPU内のベクトル処理部と同期して多重ベクトルパイプラインとして動作するスレーブCPUとに分けて設定し、前記マスターCPUは、発行元のCPUを識別する発行元CPU情報を付加した前記ベクトル処理命令を発行し、前記バスを介して自CPUを含む全てのCPUに対して転送する発行手段を備え、前記マスターCPUとスレーブCPUは、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と自CPUに設定されたマスターCPU情報とを比較し、一致する場合に転送された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、一致しない場合に前記ベクトル処理命令の無効化を行なうベクトル処理命令制御手段を備えることを特徴とする。

【0018】 請求項5の本発明では、前記マスターCPUとスレーブCPUの前記ベクトル処理命令制御手段は、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUに設定されたマスターCPU情報とを比較する比較手段と、転送された前記ベクトル処理命令を格納する命令スタックと、前記比較手段による比較結果が一致の場合、前記ベクトル処理命令を命令スタックに格納し、比較結果が不一致の場合、前記前記ベクトル処理命令を無効化する無効化処理手段と、前記ベクトル処理部のリソース状態に基づいて、前記命令ス



タックに格納された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なう命令発行処理手段とを備えることを特徴とする。

【0019】請求項6の本発明では、前記マスターCPUとスレーブCPUの前記ベクトル処理命令制御手段は、転送された前記ベクトル処理命令を前記ベクトル処理命令自体と前記発行元CPU情報に分離して出力する抽出手段と、分離された前記発行元CPU情報と、自CPUに設定されたマスターCPU情報とを比較する比較手段と、前記比較手段による比較結果が一致の場合、前記抽出手段からの前記ベクトル処理命令を命令スタックに格納し、比較結果が不一致の場合、前記前記ベクトル処理命令を無効化する無効化処理手段と、前記ベクトル処理部のリソース状態に基づいて、前記命令スタックに格納された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なう命令発行処理手段とを備えることを特徴とする。

【0020】請求項7の本発明では、前記マスターCPUとスレーブCPUの前記ベクトル処理命令制御手段は、転送された前記ベクトル処理命令を格納する命令スタックと、前記命令スタックに格納された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUに設定されたマスターCPU情報とを比較し、比較結果が一致の場合、前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、比較結果が不一致の場合、前記ベクトル処理命令に基づく命令発行を抑止し前記命令スタックの該当エリアを解放する命令発行処理手段とを備えることを特徴とする。

【0021】請求項8の本発明では、前記命令スタックには、前記発行元CPU情報を含んだ状態で前記ベクトル処理命令を格納することを特徴とする。

【0022】請求項9の本発明では、前記マスターCPUとスレーブCPUの前記ベクトル処理命令制御手段は、自CPUがスレーブCPUである場合に、自CPUの前記スカラ処理手段の動作を停止させる停止手段を備えることを特徴とする。

【0023】請求項10の本発明では、前記停止手段は、自CPUに対するマスターCPUと機能するCPUを示すマスターCPU情報と、自CPUのCPU情報を格納する格納手段と、前記ベクトル処理命令が転送された際に、前記格納手段に格納されたマスターCPU情報と自CPU情報とを比較し、不一致の場合に、自CPUの前記スカラ処理手段の動作停止信号を出力する比較手段とで構成されることを特徴とする。

【0024】請求項11の本発明では、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムにおいて、前記各CPU相互を、前記各CPUから生成するベクトル処理命令を各CPUに転送するためのパスによって接続し、前記各CPUは、発行

元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、前記ベクトルリクエストバスを介して自CPUを含む全てのCPUに対して転送する発行手段と、転送された前記ベクトル処理命令を、前記発行元CPU情報に基づいて各CPU毎に対応した複数の命令スタックに格納し、前記複数の命令スタック毎の優先順位と前記ベクトル処理手段のリソース情報に基づいて、前記ベクトル処理命令に基づく命令発行を制御するベクトル処理命令制御手段とを備えることを特徴とする。

【0025】請求項12の本発明では、前記各CPUのベクトル処理命令制御手段は、各CPU毎に対応した複数の命令スタックと、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報を検出し、前記ベクトル処理命令を対応する前記命令スタックに格納する命令発行元検出手段と、複数の前記複数の命令スタック毎に、何れの命令スタックのベクトル処理命令に基づく命令発行を優先するか決定する調停手段と、前記調停手段による決定内容と前記ベクトル処理手段のリソース情報に基づいて、前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行なう命令発行処理手段とを備えることを特徴とする。

【0026】請求項15の本発明によれば、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムの制御方法において、前記各CPUにおいて、発行元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、相互に接続されたバスを介して自CPUを含む全てのCPUに対して転送し、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と、自CPUのベクトルパイプラインに関する構成情報とを比較し、転送された前記ベクトル処理命令に対する処理内容を決定することを特徴とする。

【0027】請求項16の本発明によれば、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムの制御方法において、前記各CPUを、スカラ処理を実行すると共に、ベクトル処理命令を他のCPUに対して発行するマスターCPUと、前記マスターCPUから転送されてきたベクトル処理命令を受け取り、マスターCPU内のベクトル処理部と同期して多重ベクトルパイプラインとして動作するスレーブCPUとに分けて設定し、前記マスターCPUにおいて、発行元のCPUを識別する発行元CPU情報を付加した前記ベクトル処理命令を発行し、相互に接続されたバスを介して自CPUを含む全てのCPUに対して転送し、前記マスターCPUとスレーブCPUにおいて、転送された前記ベクトル処理命令に含まれる前記発行元CPU情報と自CPUに設定されたマスターCPU情報とを比較



し、比較結果が一致する場合に転送された前記ベクトル処理命令に基づく命令発行を前記ベクトル処理手段に対して行ない、一致しない場合に前記ベクトル処理命令の無効化を行なうことを特徴とする。

【0028】請求項20の本発明によれば、主記憶メモリを共有する複数のCPUを備え、各CPUがスカラ処理手段とベクトル処理手段を有してなる共有メモリ型ベクトル処理システムの制御方法において、前記各CPU相互を、前記各CPUから生成するベクトル処理命令を各CPUに転送するためのパスによって接続し、前記各CPUにおいて、発行元のCPUを識別する発行元CPU情報を付加したベクトル処理命令を発行し、前記パスを介して自CPUを含む全てのCPUに対して転送し、転送された前記ベクトル処理命令を、前記発行元CPU情報に基づいて各CPU毎に対応した複数の命令スタックに格納し、前記複数の命令スタック毎の優先順位と前記ベクトル処理手段のリソース情報に基づいて、前記ベクトル処理命令に基づく命令発行を制御することを特徴とする。

【0029】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照して詳細に説明する。図1は、本発明の第1の実施の形態に係るベクトル処理システム全体の構成図である。

【0030】本実施の形態に係るベクトル処理システムは、複数のCPU10a~10nを備え、これらのCPU10a~10nが単一の主記憶装置20を共有する共有メモリ型並列処理システムを構成している。各CPU10a~10nは、ベクトルリクエストバス30を介して互いに接続され、相互にベクトル処理に関するリクエストやリプライの送受信を行なうことができる。

【0031】上記各CPU10a~10nの詳細な構成を図2に基づいて説明する。

【0032】CPU10a~10nは、1つのスカラ処理部11、メモリアクセス命令制御部12、ベクトル処理命令制御部13、複数のベクトル処理部14a~14n、そしてメモリアクセスネットワーク部15から構成されている。

【0033】スカラ処理部11から外部に発行される外部処理命令「EX-RQ」は、メモリアクセス制御部12を通して、メモリアクセスネットワーク部15経由で主記憶装置20へと転送されるか、ベクトルリクエストバス30を経由して全てのCPU10a~10nのベクトル処理命令制御部13へと送られ、ベクトル処理部14a~14nへと発行される。

【0034】ここで、図3は、ベクトル処理命令制御部13の詳細な構成を示すブロック図である。

【0035】ベクトル処理命令制御部13は、2つのレジスタ42、43と、それらのレジスタ42、43の内容の比較を行なう比較器45と、ベクトルリクエストバ

ス30経由で転送されてきたベクトル処理命令の内容を分離する命令発行元情報抽出部41により得られたベクトル処理命令の発行元情報と一方のレジスタ42の内容を比較するための比較器44、命令無効化処理部46、命令スタック47、リソース管理/命令発行処理部48を有している。

【0036】命令発行元情報抽出部41から得られたベクトル処理命令と比較器44の出力は、無効化処理部46に入力された後に、命令スタック47に格納される。命令スタック47に格納されたベクトル処理命令は、ベクトル処理部14a~14nからのリソース情報等と共に、リソース管理/命令発行処理部48に入力されて、ベクトル処理部14a~14nに対して命令発行が行なわれる。

【0037】次いで、上記のように構成される第1の実施の形態によるベクトル処理システムの動作について説明する。

【0038】図2において、スカラ処理部11では命令の解読を行ないスカラ処理命令の実行処理を行なう。ここで、主記憶装置20へのアクセス命令やベクトル処理命令などといった、スカラ処理部11では実行処理できない命令が出現した場合、スカラ処理部11はこれらの命令を外部処理命令「EX-RQ」としてメモリアクセス命令制御部12へと転送する。

【0039】メモリアクセス命令制御部12では、スカラ処理部11から受け取った外部処理命令「EX-RQ」を解読し、主記憶アクセス系の命令「M-RQ」であれば、そのままメモリアクセスネットワーク部15に対して命令の発行を行なう。

【0040】一方、ベクトル処理命令であった場合には、これをベクトルリクエストバス30に送出すると共に、自身のベクトル処理命令制御部13に対してもベクトルリクエストバス30を経由して発行を行なう。

【0041】ベクトル処理命令制御部13では、メモリアクセス命令制御部12から送られてきた自CPUが発行したベクトル処理命令と他のCPUからベクトルリクエストバス30を経由して転送されてきたベクトル処理命令とを受け取り、自CPU内のベクトル処理部14a~14nに対してそのリソース状態を管理しながら命令を発行する。

【0042】メモリアクセスネットワーク部15は、メモリアクセス命令制御部12からの主記憶アクセス命令「M-RQ」を受け取り、主記憶装置20に対して命令を発行すると共に、主記憶装置20からの読み出しデータを受け取り、命令種別に応じてデータをスカラ処理部11もしくはベクトル処理部14a~14nに対して戻す。

【0043】次で、図3と図4のフローチャートを参照して、ベクトル処理命令制御部13の動作について述べる。

【0044】ベクトル処理命令制御部13に送られてきたベクトル処理命令は、命令発行元情報抽出部41にて命令を送出したCPUに関する情報とベクトル処理命令本体とに分離される（ステップ401）。

【0045】ベクトル処理命令制御部13には、自CPUに対してマスターとして外部より設定されたCPUの番号を記憶しているレジスタ42と、自CPU番号を記憶しているレジスタ43とが備えられている。これら2つのレジスタ42、43に対しては、システムの起動前に初期化動作として上記番号をそれぞれ設定しておくものとする。

【0046】本実施の形態では、共有メモリ型並列処理システムの各CPU10a~10nをマスターCPUとスレーブCPUとに分けて設定する。マスターCPUはスカラ処理を実行すると共に、ベクトル処理命令を他のCPUに対して発行することができる。これに対してスレーブCPUは、マスターCPUから転送されてきたベクトル処理命令を受け取り、マスターCPU内のベクトル処理部14a~14nと同期して多重ベクトルパイプラインとして動作することになる。この時、スレーブCPUのスカラ処理部11は休止状態として、そのベクトル処理部14a~14nとベクトル処理命令制御部13、及びメモリアクセスネットワーク部15のみが有効に機能することになる。

【0047】マスターCPU番号を格納したレジスタ42と自CPU番号を格納したレジスタ43の内容は、比較器45で比較され（ステップ402）、不一致の場合には、自CPUがスレーブCPUであると判断され、自CPUのスカラ処理部11に対して動作を停止するよう制御する（ステップ403）。

【0048】一方、ベクトルリクエストバス30を通して転送されたベクトル処理命令から命令発行元情報抽出部41において取り出された命令発行元CPU番号と、マスターCPU番号を格納しているレジスタ42の内容とが、もう1つの比較器44で比較される（ステップ404）。この際の比較結果と、命令発行元情報抽出部41にて分離されたベクトル処理命令は、無効化処理部46に入力される（ステップ405）。

【0049】比較器44による比較結果が不一致となっていた場合（ステップ406）、入力されたベクトル処理命令はスレーブとして動作している自CPUのマスターCPUから発行されたベクトル処理命令ではないため、無効化処理部46にて無効化される（ステップ407）。具体的には、比較器44による比較結果に応じて、ベクトル処理命令に有効又は無効を示すフラグを付し、無効化処理部46では、そのフラグによって有効なベクトル処理命令のみを命令スタック47に格納する。無効なベクトル処理命令は命令スタック47に格納しない。

【0050】当然のことながら、自CPUがマスターC

PUとして動作しており、転送されてきたベクトル処理命令が自CPUの発行した命令であれば、比較器44の結果は一致を示すので無効化されることはない。

【0051】無効化処理部46で無効化されなかったベクトル処理命令は、自CPU内のベクトル処理部14a~14nで処理すべき命令であることから、命令スタック47に受付順に格納される（ステップ408）。無効化されたベクトル処理命令は、命令スタック47に格納されずに破棄される。

【0052】リソース管理／命令発行制御部48では、自CPU内のベクトル処理部のリソース14a~14nを管理している。命令スタック47に格納された命令は、このリソース管理／命令発行処理部48において優先順位、ならびにベクトル処理部14a~14nのリソース状況に応じて、発行が可能な順に自CPUのベクトル処理部14a~14nに対して命令発行が行なわれる（ステップ409）。ここでは、命令スタック47への格納順には従わずに命令の追い越し発行も可能である。

【0053】なお、各スレーブCPUにおけるベクトル処理が終了した時点で、処理の終了がマスターCPUに通知される。マスターCPUでは、全てのスレーブCPUからの終了通知を受け取ったことを確認した上で次のベクトル処理命令の発行を行なうことになる。

【0054】以上により、マスターCPUに設定されたCPUと、このCPU番号をマスターCPU番号として記憶している複数のCPUは、一体となって動作する多重ベクトルパイプラインのプロセッサと見なすことができるようになる。

【0055】この時、マスターCPUのスカラ処理部11だけが機能しており、スレーブCPUのスカラ処理部11はベクトル処理命令制御部13からの制御信号「HALT」により機能が停止されている。

【0056】マスターCPUのスカラ処理部11から発行されたベクトル処理命令は、ベクトルリクエストバス30を通して自CPUを含めてスレーブCPUの各ベクトル処理命令制御部13において有効と判断され、これら複数のCPUにおけるベクトル処理部14a~14nにおいて並列動作によって処理されることになる。

【0057】仮に、1つのCPUに1つのベクトル処理部が存在する場合、主記憶装置20を共有しているCPU数を32とすると、通常は「1スカラ処理部+1ベクトル処理部」のCPUが32個存在するシステムとして固定化されてしまうが、1つのマスターCPUに1つのスレーブCPUを対応させた場合、「1スカラ処理部+2ベクトル処理部」のCPUが16個存在するシステムのように動作させることが可能となる。

【0058】また、マスターCPUとスレーブCPUの設定内容により、「1スカラ処理部+1ベクトル処理部」のCPUと「1スカラ処理部+4ベクトル処理部」のCPUを1つのシステムの中に混在させるような構成

も取ることができる。すなわち、マスターCPUとスレーブCPUの設定内容により、種々の構成を構築できるようになる。

【0059】次いで、第2の実施の形態に係るベクトル処理システムについて説明する。

【0060】図5は、第2の実施の形態によるベクトル処理システムのベクトル処理命令制御部13の構成を示すブロック図である。ベクトル処理命令制御部13以外の構成については、上述の第1の実施の形態と同様である。

【0061】図5に示すベクトル処理命令制御部13では、マスターCPU番号を記憶しているレジスタ42aと自CPU番号を記憶しているレジスタ43aを有しており、この2つのレジスタ42a、43aの内容を比較する比較器44aの出力に基づいて自CPUのスカラ処理部11の機能を停止させることについては、図3の構成の場合と同じである。また、このベクトル処理命令制御部13には、レジスタ42aのマスターCPU番号とベクトル処理命令に含まれる発行元CPU情報とを比較する機能と、比較結果に基づいて命令発行を制御する機能を有するリソース管理/命令発行制御部48aが備えられている。

【0062】このベクトル処理命令制御部13の動作を図6のフローチャートを参照して説明する。この構成例では、図6に示すように、ベクトルリクエストバス30を通して転送されてきたベクトル処理命令は、何も処理をせずそのまま命令スタック47aに順番に格納される(ステップ601)。従って、命令スタック47aには、ベクトル処理命令と共に、発行元CPU情報を格納しておくレコードが設けられている。

【0063】また、マスターCPU番号を格納したレジスタ42aと自CPU番号を格納したレジスタ43aの内容は、比較器45aで比較され(ステップ602)、不一致の場合には、自CPUがスレーブCPUであると判断され、自CPUのスカラ処理部11に対して動作を停止するよう制御する(ステップ603)。

【0064】次で、リソース管理/命令発行制御部48aでは、ベクトル処理部への命令発行を行なう際に、ベクトル処理命令に付随している発行元CPU情報と、マスターCPU番号を記憶しているレジスタ42aの内容を比較することにより(ステップ604)、番号が不一致である場合には不適切なベクトル処理命令の発行を抑制すると共に、命令スタック47a中の該当エリアを解放する(ステップ605)。すなわち、命令スタック47aに格納する前に無効化処理を行なうのではなく、実際に命令発行を行なう際に無効化処理を行なうように構成している。

【0065】また、番号が一致する場合には、第1の実施の形態におけるリソース管理/命令発行制御部48と同様に、命令スタック47aに格納された命令が、優先

順位、ならびにベクトル処理部14a~14nのリソース状況に応じて、発行が可能な順に自CPUのベクトル処理部14a~14nに対して命令発行が行なわれる(ステップ606)。

【0066】上述した第1の実施の形態においては、ベクトル処理命令の命令発行元CPU番号を抽出する命令発行元情報抽出部41、命令発行元CPU番号とマスターCPU番号を比較する比較器44及び比較結果によってベクトル処理命令の無効化を行なう無効化処理部46を備えることで命令スタック47に無効なベクトル処理命令を格納しない構成としたのに対して、この第2の実施の形態では、送られた発行元CPU情報を含むベクトル処理命令を全て命令スタック47aに格納し、リソース管理命令発行処理部48aによる命令発行処理の段階で、適切なベクトル処理命令のみを発行し、不適切なベクトル処理命令について命令スタック47aのエリアを解放する構成としている。従って、第1の実施の形態と第2の実施の形態とを比較した場合、ハードウェア量については第2に実施の形態の方が少なく済み、命令スタックの記憶容量については第1の実施の形態の方が小さくすることができる。

【0067】一方、よりスカラ性能を重視したシステム構成として複数の独立したスカラ処理部から共有される多重ベクトルパイプラインという構成も可能である。すなわち、複数のプロセッサ中に存在するベクトルパイプラインを全てまとめて1つの多重ベクトルパイプラインと見なし、独立する各プロセッサのスカラ処理部から単一のベクトルパイプラインを共有しているように動作するシステム構成である。

【0068】これを実現した第3の実施の形態に係るベクトル処理システムのベクトル処理命令制御部13の構成を図7に示す。なお、ベクトル処理命令制御部13以外の構成については、上述した第1の実施の形態と同一であるので共通の符号を付して説明を省略する。

【0069】第3の実施の形態に係るベクトル処理システムにおいて、ベクトル処理命令制御部13は、命令発行元検出部61と、各CPU毎に設けられた命令スタック63a~63nと、命令スタック63a~63nに設定された優先順位に基づいて発行順の調停を行なう調停部62と、リソース管理/命令発行処理部64とで構成される。

【0070】以下、本実施の形態によるベクトル処理命令制御部13の動作を図8のフローチャートを参照して説明する。

【0071】ベクトルリクエストバス30を通して転送されてくるベクトル処理命令は、命令発行元検出部61を経由して各CPU毎に設けられた命令スタック63a~63nに格納される。格納されたベクトル処理命令は、調停部62による調停結果と各ベクトル処理部14a~14nからのリソース情報「V-RP」と共に、リ

ソース管理／命令発行制御部 64 に入力された後に各ベクトル処理部 14a～14n に対して発行される。

【0072】ここでは、ベクトルリクエストバス 30 を通して転送されてきたベクトル処理命令は、命令発行元検出部 61 で発行元の CPU 番号が検査される（ステップ 801）。その後、ベクトル処理命令は、発行元 CPU 毎に設けられた命令スタック 63a～63n に分かれて格納される（ステップ 802）。

【0073】そして、何れかの命令スタック 63a～63n から命令を発行するかどうかを、優先順位によって競合調停する調停部 62 により決定する（ステップ 803）。調停部 62 は、例えばラウンドロビン方式によって何れかの命令スタック 63a～63n から命令を発行するかを決定する。この調停部 62 の出力と、各ベクトル処理部のリソース情報とを用いて、リソース管理／命令発行制御部 64 で発行命令を決定する（ステップ 804）。

【0074】この際、発行元 CPU が同一のベクトル処理命令については、命令スタックへの格納順序を越えて追い越し発行することはできないが、発行元 CPU が異なっていればリソースの状況によって追い越し発行を行ってもデータ競合は起こらないため問題にはならない。従って、各命令スタック間での格納順序については、特に記憶しておく必要はない。また、発行元 CPU が同一のベクトル処理命令に関しても、アクセスアドレスを比較することで同一アドレスに対するアクセスを回避するための相応のリソース管理手段を準備すれば、追い越し発行が可能である。

【0075】以上により、各 CPU から発行されたベクトル処理命令は全ての CPU のベクトル処理命令制御部に転送され発行処理が行なわれることになる。この時ベクトル処理命令の発行元 CPU 別に管理されるため、各 CPU に存在するベクトル処理部は全 CPU で統合された単一のベクトル処理部として、全 CPU のスカラ処理部から共有されているように動作することになる。

【0076】上述したベクトル処理システムは、ハードウェア的に実現することは勿論として、図 2 に示すように、磁気ディスク、半導体メモリその他の記録媒体 18 に記録された、上述した各機能を実現するための制御プログラムによってソフトウェア的に実現することも可能である。この制御プログラムは、記録媒体 18 から CPU に読み込まれ、CPU の動作を制御することにより、上述したベクトル処理命令制御の機能を実現する。すなわち、図 4、図 6 及び図 8 に示される処理を実行する。

【0077】なお、本発明は上述した実施の形態に限定されるものではなく、その技術思想の範囲内において様々に変形して実施することができる。例えば、図 1 におけるシステム全体構成図において、各 CPU 10a～10n 間でベクトル処理命令を転送するためのベクトルリクエストバス 30 は、単一バスとして記述されている。

しかしながら、この転送手段は単一バスに限定されるものではなく、多重バスやクロスバススイッチなどあらゆる接続手段によって実現できることは明らかである。

【0078】

【発明の効果】以上説明したように本発明のベクトル処理システムとその制御方法によれば、以下に述べるような効果が得られる。

【0079】第 1 に、マスタとなるプロセッサとスレーブとなるプロセッサとの割り当て方法を変更することにより、スカラ処理部に付随するベクトルパイプライン数を用途に応じて柔軟に変更できる構成としたので、ベクトル化率やベクトル長の異なる様々なアプリケーションに対して最適な多重ベクトルパイプライン構成を有するベクトル処理システムを提供することができる。

【0080】これは、各 CPU から発行されるベクトル処理命令を全ての CPU に転送するバスを設け、各 CPU のベクトル処理命令制御手段で転送されてきたベクトル処理命令とその発行元 CPU 情報を読みとり、自 CPU がスレーブ状態にある場合に記憶しているマスター CPU 番号と発行元 CPU 番号が一致した場合にだけ転送されてきたベクトル処理命令を処理することができるためである。

【0081】これにより、並列処理システム中の各 CPU の中からマスター CPU とスレーブ CPU の組み合わせを変更することで、例えば、各 CPU を全てマスター CPU として自 CPU のベクトル処理命令を自 CPU だけで処理する最もスカラスループット性能を重視した構成から、1 つのマスター CPU に対して残りの全ての CPU を全てスレーブ CPU として動作される大規模でかつベクトル長が極めて長い処理向きの構成まで、柔軟にシステムを変更できる。

【0082】第 2 に、前記第 1 の効果の延長線上にあるシステム形態として、よりスカラ処理性能を重視したアプリケーション向けに、複数のスカラ処理手段から共有される単一のベクトル処理手段を有した共有メモリ型並列処理システムを提供することができる。

【0083】これは、各 CPU のベクトル処理命令制御に CPU 毎のベクトル処理命令スタックを準備し、各 CPU 間で転送されてきたベクトル処理命令をその発行元 CPU 毎に分類して命令スタックに格納して、各命令スタック内の命令の競合を調停しながら順次ベクトル処理部に対してベクトル処理命令を発行することで、全 CPU 中に存在するベクトル処理手段があたかも単一のベクトル処理手段として全ての CPU から共有されているように動作することが出来るためである。

【0084】これにより、ベクトル処理命令の出現頻度が極めて低いスカラスループット性能重視のアプリケーション分野に対しても、ベクトル処理リソースを有効利用してより効率の良い処理が可能なシステムを提供することができる。

【0085】第3に、スカラ処理手段とベクトル処理手段とを1チップに集積化したLSIを開発することが可能になり、開発工数やコストを軽減することができる。

【0086】これは、スカラ処理手段に対する多重ベクトルパイプラインの構成を外部からの設定により柔軟に変更できるようになったために、これまでは困難であったスカラ処理手段とベクトル処理手段を同一のLSIに集積化することが可能になり、その結果LSIの開発品種数が削減できることによる。

【図面の簡単な説明】

【図1】 本発明の第1の実施の形態に係るベクトル処理システム全体の構成図である。

【図2】 上記第1の実施の形態に係るベクトル処理システムの各CPUの詳細な構成を示すブロック図である。

【図3】 上記第1の実施の形態に係るベクトル処理システムのベクトル処理命令制御部の詳細を示すブロック図である。

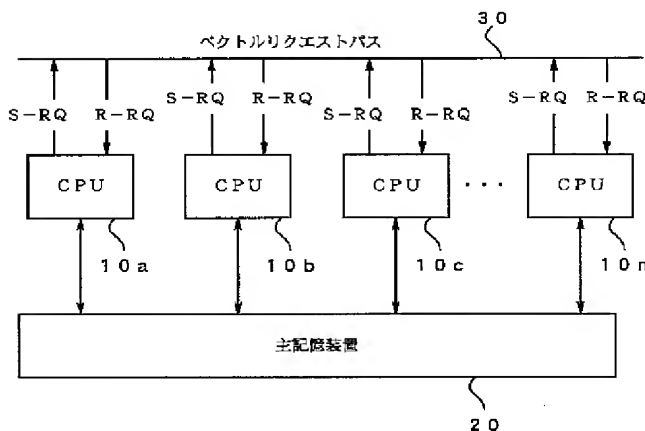
【図4】 上記第1の実施の形態に係るベクトル処理システムのベクトル処理命令制御部の動作を説明するフローチャートである。

【図5】 第2の実施の形態に係るベクトル処理システムのベクトル処理命令制御部の構成を示すブロック図である。

【図6】 上記第2の実施の形態に係るベクトル処理システムのベクトル処理命令制御部の動作を説明するフローチャートである。

【図7】 第3の実施の形態に係るベクトル処理システムのベクトル処理命令制御部の構成を示すブロック図で

【図1】



ある。

【図8】 上記第3の実施の形態に係るベクトル処理システムのベクトル処理命令制御部の動作を説明するフローチャートである。

【図9】 従来のベクトル処理装置におけるCPUを用いた共有メモリ型並列処理システムの構成を示すブロック図である。

【図10】 図9に示すベクトル処理装置の各CPUの構成を示すブロック図である。

10 【符号の説明】

10a~10n CPU

11 スカラ処理部

12 メモリアクセス命令制御部

13 ベクトル処理命令制御部

14a~14n ベクトル処理部

15 メモリアクセスネットワーク部

20 主記憶装置

30 ベクトルリクエストバス

41 命令発行元情報抽出部

20 42, 43, 42a, 43a レジスタ

44, 45, 44a 比較器

46 無効化処理部

47, 47a 命令スタック

48, 48a リソース管理命令発行処理部

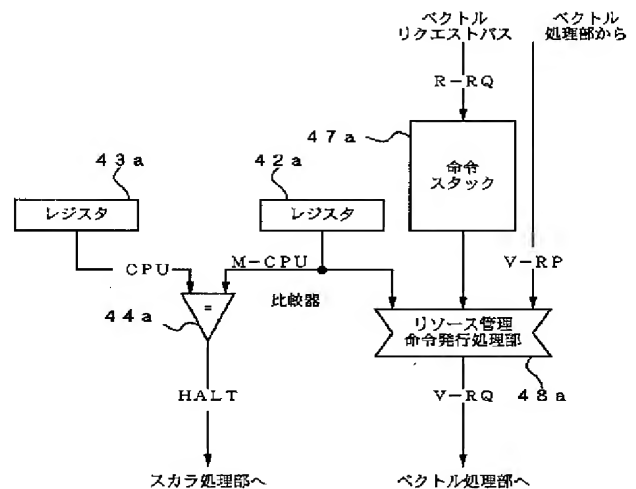
61 命令発行元検出

62 調停部

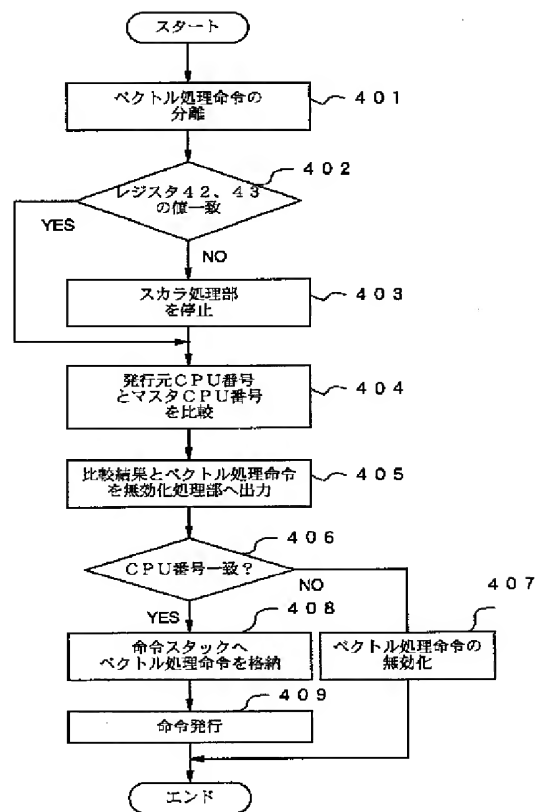
63a~63n 命令スタック

64 リソース管理/命令発行処理部

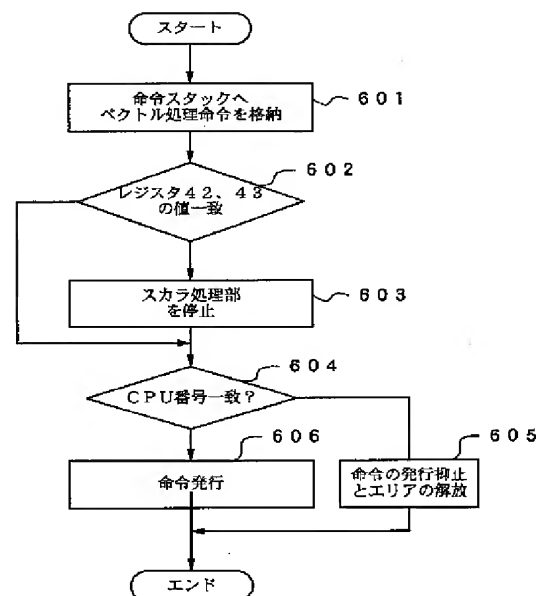
【図5】



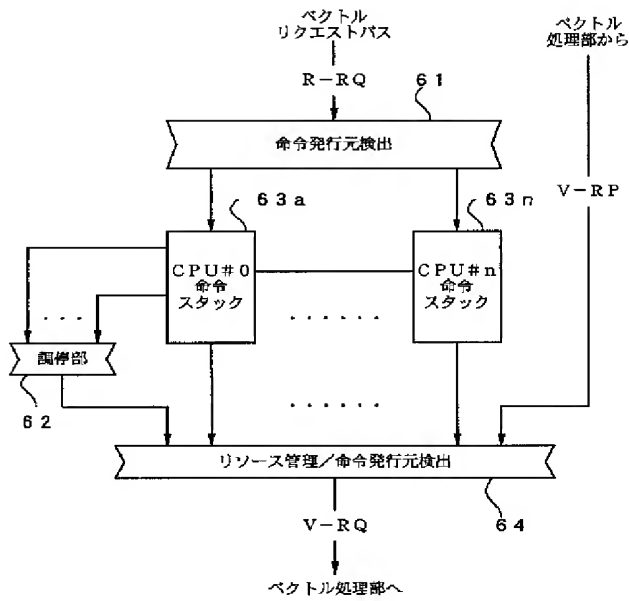
【图 4】



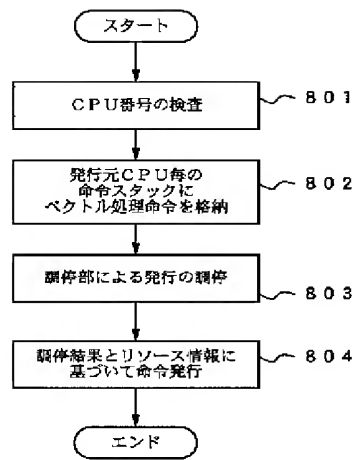
【図 6】



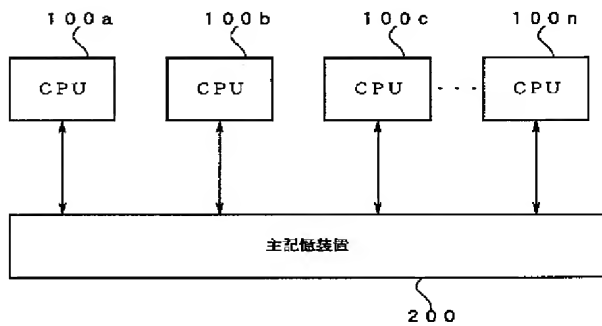
【図7】



【図8】



【図9】



【図10】

